Class 6: R Functions

Lana (PID: A17013518)

2024-01-25

R Functions

Functions are how we get stuff done. We call functions to do everything useful in R.

One cool thing about R is that it makes writing your own functions comparatively easy.

All functions in R have at least three things:

- A **name** (we get to pick this)
- One or more **input arguments** (the input to our function)
- The **body** (lines of code that do the work)

```
funname <- function(input 1, input 2) {
  # The body with R code
}
```

Let's write a silly function to add two numbers:

```
x <- 5
y <- 1
x + y
```

```
[1] 6
```

```
addme <- function(x, y=1) {
  x + y
}
```

```
addme(100,100)
```

```
[1] 200
```

Lab for today

Write a function to grade student work from class.

Start with a simplified version of the problem:

```
student1 <- c(100, 73, 100, 88, 79)
student2 <- c(85, 64, 78, 89, 78)
student3 <- c(83, 69, 77, 100, 77)
student4 <- c(88, NA, 73, 100, 76)
```

We want to drop the lowest score before getting the mean:

```
which.min(student1)
```

```
[1] 2
```

```
which.min(student2)
```

```
[1] 2
```

Cool - it is the 2nd element of the vector that has the lowest score. Can I remove this one?

```
student1[which.min(student1)]
```

```
[1] 73
```

```
student2[which.min(student2)]
```

```
[1] 64
```

We can use the three minus trick for extracting.

```
x <- 1:5
x[-3]
```

```
[1] 1 2 4 5
```

Let's find the average

```
mean(student1)
```

```
[1] 88
```

```
mean(student2, na.rm = TRUE)
```

```
[1] 78.8
```

```
# Find the lowest score
ind <- which.min(student1)
# remove the lowest score and find mean
mean(student1[-ind])
```

```
[1] 91.75
```

Use a common shortcut and use `x` as my input

```
x <- student1
```

How to make NA work:

```
!c(F,F,F)
```

```
[1] TRUE TRUE TRUE
```

```
# y[ is.na(y)
```

Okay let's solve this:

```
# first set a variable to a student
x <- student1
# next, set value of NA to 0
x[ is.na(x)] <- 0
# locate and remove the min value and calculate mean
mean( x[-which.min(x)])
```

```
[1] 91.75
```

Last step now that I have my working snippet is to make my function:

```
grade <- function(x) {
  x[ is.na(x)] <- 0
  mean( x[-which.min(x)])
}
```

Q1. Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single score. If a student misses a homework (i.e. has an NA value) this can be used as a score to be potentially dropped. Your final function should be adquately explained with code comments and be able to work on an example class gradebook such as this one in CSV format: "https://tinyurl.com/gradeinput"

```
url <- "https://tinyurl.com/gradeinput"
gradebook <- read.csv(url, row.names = 1)

head(gradebook)
```

```
          hw1 hw2 hw3 hw4 hw5
student-1 100  73 100  88  79
student-2  85  64  78  89  78
student-3  83  69  77 100  77
student-4  88  NA  73 100  76
student-5  88 100  75  86  79
student-6  89  78 100  89  77
```

```
results <- apply(gradebook, 1, grade)
results
```

```
 student-1  student-2  student-3  student-4  student-5  student-6  student-7
     91.75      82.50      84.25      84.25      88.25      89.00      94.00
 student-8  student-9 student-10 student-11 student-12 student-13 student-14
     93.75      87.75      79.00      86.00      91.75      92.25      87.75
student-15 student-16 student-17 student-18 student-19 student-20
     78.75      89.50      88.00      94.50      82.75      82.75
```

```
grade <- function(x) {
  x[ is.na(x)] <- 0
  mean( x[-which.min(x)])
}
```

Q2. Using your grade() function and the supplied gradebook, Who is the top scoring student overall in the gradebook?

```
max(results)
```

```
[1] 94.5
```

```
which.max(results)
```

```
student-18
        18
```

Q3. From your analysis of the gradebook, which homework was toughest on students (i.e. obtained the lowest scores overall?

```
results <- apply(gradebook, 2, mean, na.rm = T)
results
```

```
     hw1      hw2      hw3      hw4      hw5
89.00000 80.88889 80.80000 89.63158 83.42105
```

```
min(results)
```

```
[1] 80.8
```

```
which.min(results)
```

```
hw3
  3
```

Q4. Optional Extension: From your analysis of the gradebook, which homework was most predictive of overall score (i.e. highest correlation with average grade score)?

```
# make all NA to 0
mask <- gradebook
mask[is.na(mask)] <- 0
```

We can use the `cor()` function for correlation analysis.

```r
results <- apply(gradebook, 1, grade)
cor(mask$hw5, results)
```

```
[1] 0.6325982
```

I need to use this `apply()` function to use this for the entire course.

```r
apply(mask, 2, cor, results)
```

```
      hw1       hw2       hw3       hw4       hw5 
0.4250204 0.1767780 0.3042561 0.3810884 0.6325982 
```